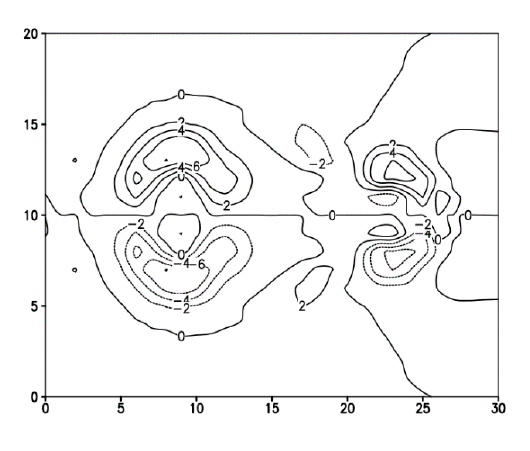
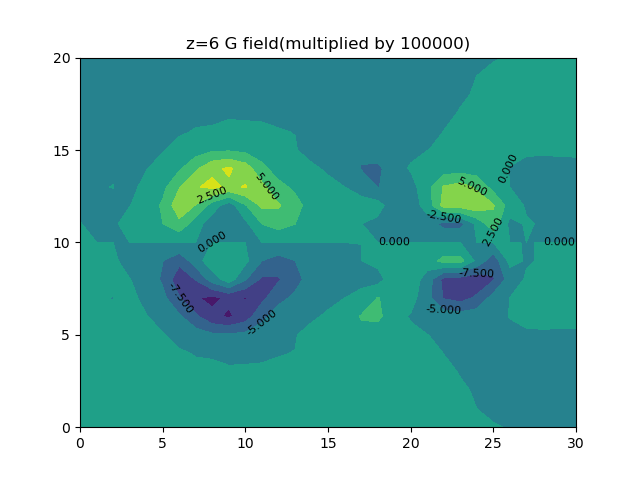
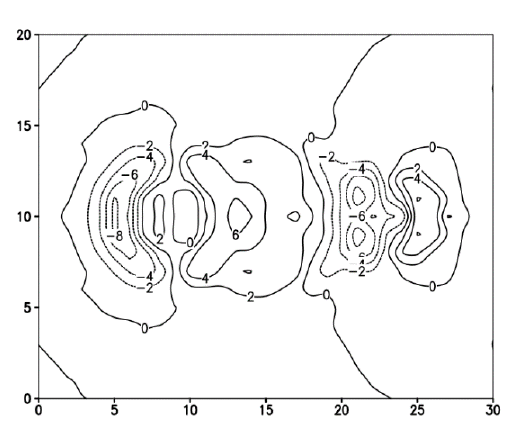
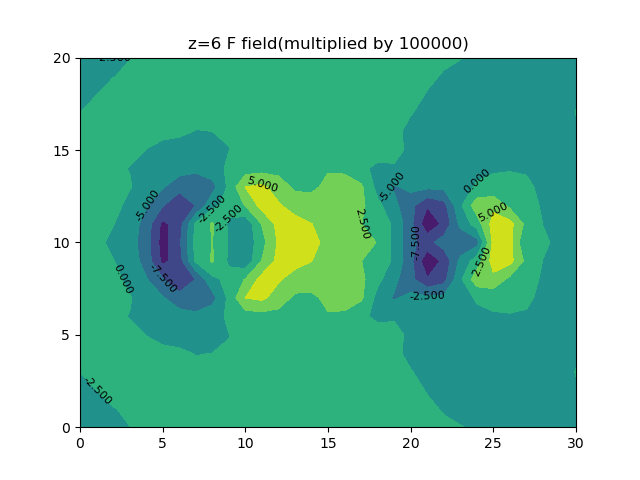
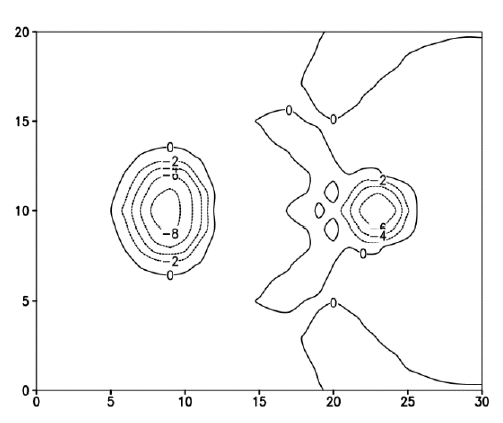
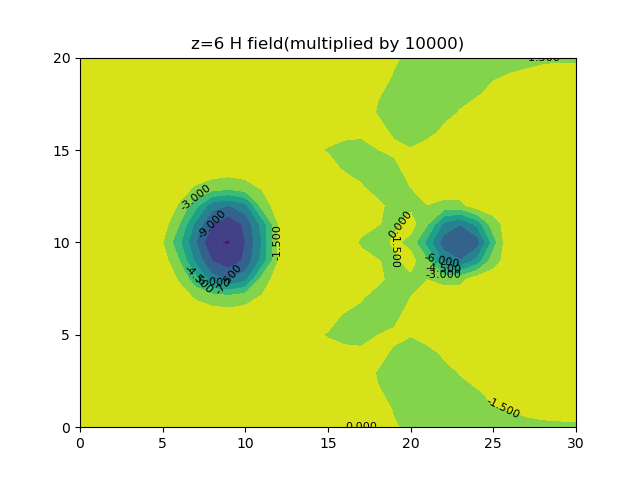
高等應用數學 HW6 黃展皇110621013

1. Plot the fields of F、G and H(z=6)：





可以看到繪製出來的圖對比作業附圖來說都是對的，而且更漂亮😊

1. retrieve pressure and temperature perturbation fields (𝜋′,𝜃𝑐′) at each layer.

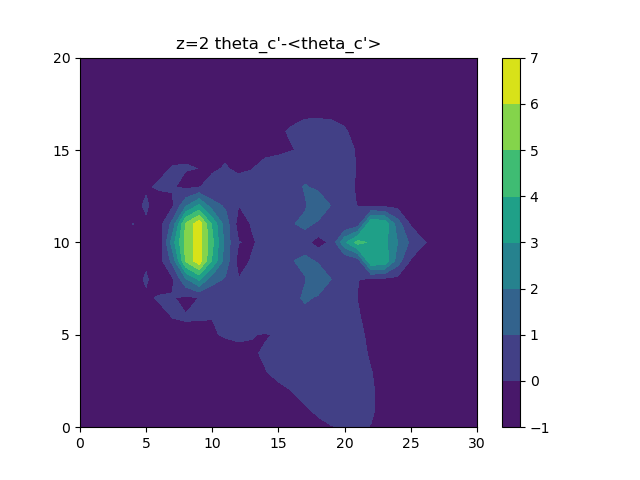
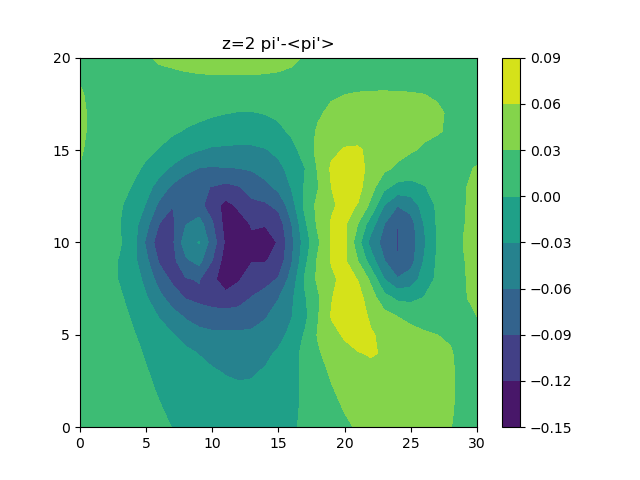
作法：𝜋′反演採用SOR，eps=1.5，結束條件是每一點相對誤差均小於0.001，輸入該層F、G即可得到該層𝜋′。pi矩陣代表𝜋′一開始都是0.0，定義Neumann邊界條件使用中插法、第一內圈F、G以及第二內圈pi得到外圈pi，並進行內圈SOR：

sor\_result = -0.25\*dx\*dy\*(median\_interpolation(F[y, x-1], F[y, x+1], dx) + median\_interpolation(G[y-1, x], G[y+1, x], dy)) + (pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4

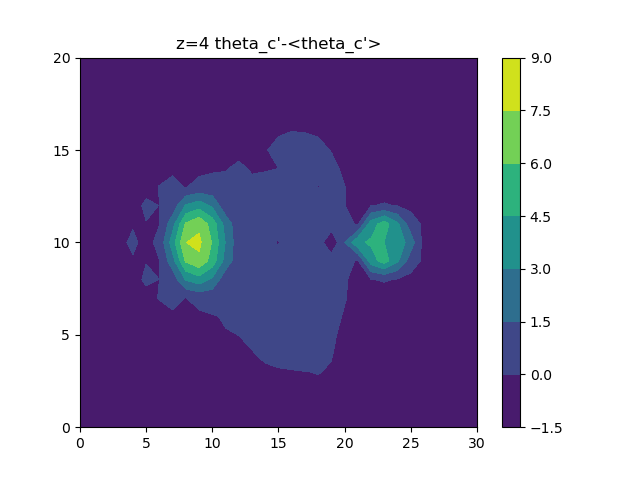
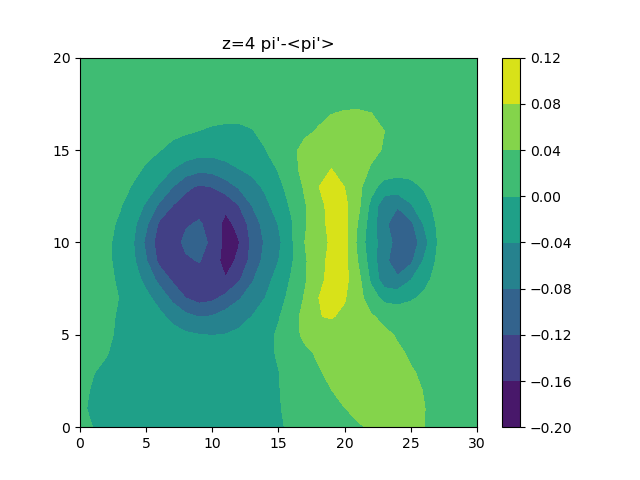
pi[y, x] += eps \* (sor\_result - pi[y, x])，重複以上步驟迭代即可得到𝜋′場。

𝜃𝑐′反演則使用該層上下的𝜋′場以及該層H場還有theta\_0\_avg, theta\_v0\_avg，套用theta\_c = (dpidz-H)\*theta\_0\_avg\*theta\_v0\_avg/g計算即可知道𝜃𝑐′場。

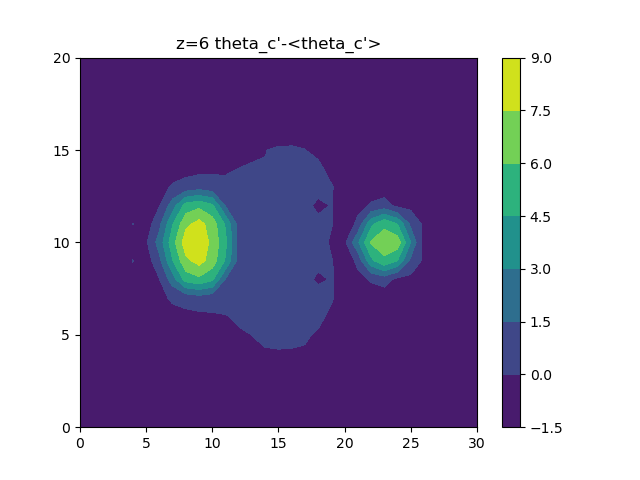
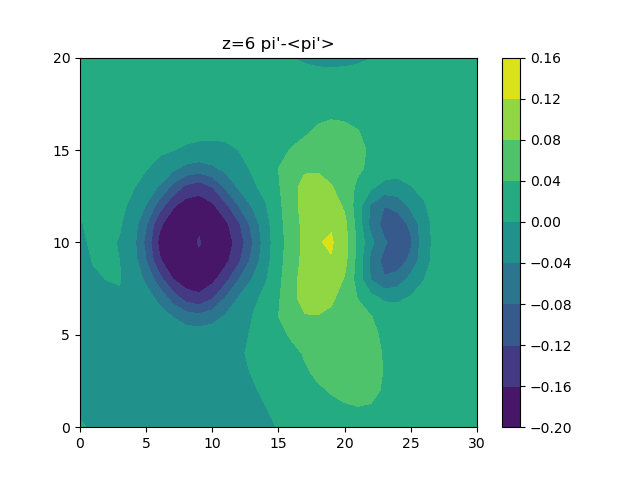
(z=2) 𝜋′-<𝜋′>場 𝜃𝑐′-<𝜃𝑐′>場



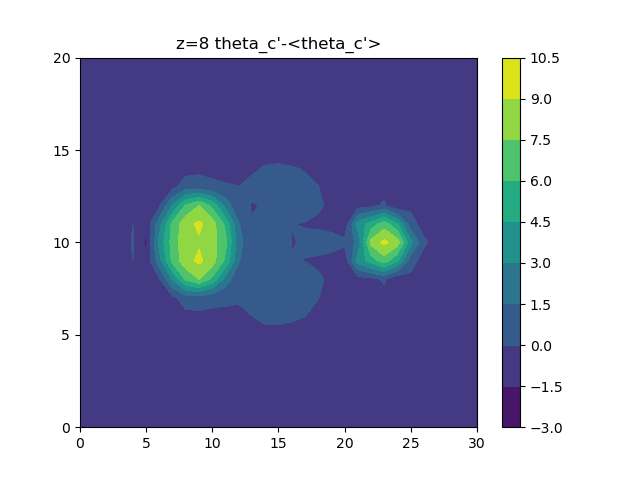
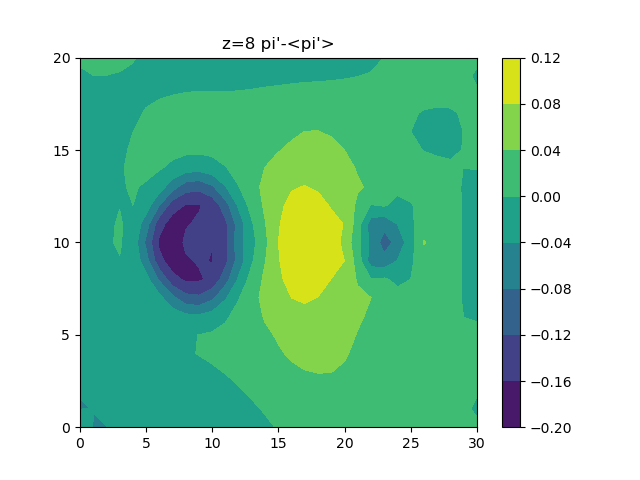
(z=4) 𝜋′-<𝜋′>場 𝜃𝑐′-<𝜃𝑐′>場



(z=6) 𝜋′-<𝜋′>場 𝜃𝑐′-<𝜃𝑐′>場



(z=8) 𝜋′-<𝜋′>場 𝜃𝑐′-<𝜃𝑐′>場



可以看到反演出來的𝜋′-< 𝜋′>場在中心有相對高值，並在左右各有相對低值，又以左側為更低；而𝜃𝑐′-<𝜃𝑐′>場則是呈現左右兩個明顯的熱胞柱。

原始碼：(環境：win10、conda4.10.3、python3.8.8、numpy1.19.5)

import os

import numpy as np

import matplotlib.pyplot as plt

x\_n, y\_n, z\_n = 31, 21, 11

dx, dy, dz = 1000, 1000, 250

# read hw6 data, 31\*21\*11\*5(x,y,z,var), z:1~11

def read\_hw6\_data(z, parameter):

    parameter\_list = ['ff', 'gg', 'hh', 'theta0', 'thetav0', 'ans\_thetac']

    data = np.fromfile('homework6.bin', dtype='<f4', count=-1, sep='')

    init\_lndex = x\_n\*y\_n\*z\_n\*parameter\_list.index(parameter) + x\_n\*y\_n\*(z-1)

    return np.reshape(data[init\_lndex:init\_lndex+x\_n\*y\_n], (y\_n, x\_n))

# plot parameter

def plot\_parameter(plane\_data, title, scale):

    plane\_data\_new = np.array(plane\_data)

    assert id(plane\_data) != id(plane\_data\_new)

    plane\_data\_new \*= scale

    C = plt.contourf(plane\_data\_new) #, levels=[i for i in range(-10, 10, 2)]

    plt.clabel(C, fontsize=8, colors='black', inline=False)

    plt.title(title)

    plt.xticks(range(0, 31, 5))

    plt.yticks(range(0, 21, 5))

    plt.savefig('6-'+title)

    #plt.show()

    plt.close()

# Input pre, post, and d and output interpolation differential.

def median\_interpolation(front, behind, d):

    return (behind-front)/(2\*d)

def cal\_pi(F, G, z\_name):

    # if pi has been cal

    filepath = os.path.join('.', '6-'+z\_name+'\_pi.npy')

    if os.path.isfile(filepath):

        pi = np.load(os.path.join('.', '6-'+z\_name+'\_pi.npy'))

    else:

        # pi B.C. init

        pi = np.zeros((y\_n, x\_n), dtype=np.float64)

        for y in range(1, y\_n-1):

            pi[y, 0] = -2\*dx\*F[y, 1] + pi[y, 2]

            pi[y, x\_n-1] = 2\*dx\*F[y, x\_n-2] - pi[y, x\_n-3]

        for x in range(1, x\_n-1):

            pi[0, x] = -2\*dy\*F[1, x] + pi[2, x]

            pi[y\_n-1, x] = 2\*dy\*F[y\_n-2, x] - pi[y\_n-3, x]

        # SOR + B.C. renew

        count = 0

        eps = np.array([1.5], dtype=np.float64)[0]

        threshold = np.array([10\*\*-3], dtype=np.float64)[0]

        while True:

            old\_pi = np.array(pi) # call by values

            count += 1

            for y in range(1, y\_n-1):

                for x in range(1, x\_n-1):

                    #sor\_result = 2\*dx\*dy\*(median\_interpolation(F[y, x-1], F[y, x+1], dx) + median\_interpolation(G[y-1, x], G[y+1, x], dy)) + (pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4

                    sor\_result = -0.25\*dx\*dy\*(median\_interpolation(F[y, x-1], F[y, x+1], dx) + median\_interpolation(G[y-1, x], G[y+1, x], dy)) + (pi[y+1, x]+pi[y-1, x]+pi[y, x+1]+pi[y, x-1])/4

                    pi[y, x] += eps \* (sor\_result - pi[y, x])

            skip\_flag = False

            max\_rela\_e = np.array([0.0], dtype=np.float64)[0]

            for y in range(1, y\_n-1):

                for x in range(1, x\_n-1):

                    if old\_pi[y, x] == np.array([0.0], dtype=np.float64)[0]:

                        max\_rela\_e = 10\*threshold # abs fail

                        skip\_flag = True

                    if skip\_flag == True:

                        break

                    max\_rela\_e = max(max\_rela\_e, abs((pi[y, x]-old\_pi[y, x])/old\_pi[y, x]))

                if skip\_flag == True:

                    break

            ''' MSE

            skip\_flag = False

            mse = np.array([0.0], dtype=np.float64)[0]

            for i in range(2, n-2):

                for j in range(2, n-2):

                    if old\_lamb[i][j] == np.array([0.0], dtype=np.float64)[0]:

                        mse = 10\*mse\_threshold # abs fail

                        skip\_flag = True

                    if skip\_flag == True:

                        break

                    mse += (lamb[i][j]-old\_lamb[i][j])\*\*2

                if skip\_flag == True:

                    break

            '''

            # B.C. renew

            for y in range(1, y\_n-1):

                pi[y, 0] = -2\*dx\*F[y, 1] + pi[y, 2]

                pi[y, x\_n-1] = 2\*dx\*F[y, x\_n-2] - pi[y, x\_n-3]

            for x in range(1, x\_n-1):

                pi[0, x] = -2\*dy\*G[1, x] + pi[2, x]

                pi[y\_n-1, x] = 2\*dy\*G[y\_n-2, x] - pi[y\_n-3, x]

            if skip\_flag or count%100==0:

                print(count, skip\_flag, max\_rela\_e)

            if max\_rela\_e < threshold:

                break

            #if count == 4000:

            #    break

            #if count % 1000 == 0:

            #    plt.contourf(pi)

            #    plt.colorbar()

            #    plt.show()

        # -c

        pi\_avg = np.mean(pi[0:y\_n, 0:x\_n])

        print('pi\_avg:', pi\_avg)

        pi -= pi\_avg

        np.save('6-'+z\_name+'\_pi.npy', pi)

    plt.contourf(pi)

    plt.colorbar()

    plt.title(z\_name+" pi'-<pi'>")

    plt.xticks(range(0, 31, 5))

    plt.yticks(range(0, 21, 5))

    plt.savefig('6-'+z\_name+'\_pi')

    #plt.show()

    plt.close()

def cal\_theta\_c(H, theta\_0, theta\_v0, z\_name):

    # if pi has been cal

    filepath = os.path.join('.', '6-'+z\_name+'\_theta\_c.npy')

    if os.path.isfile(filepath):

        theta\_c = np.load(os.path.join('.', '6-'+z\_name+'\_theta\_c.npy'))

    else:

        assert z\_name in ['z='+str(i) for i in range(2, 11)]

        middle\_num = int(z\_name.split('=')[-1])

        under = np.load('6-z='+str(middle\_num-1)+'\_pi.npy')

        up = np.load('6-z='+str(middle\_num+1)+'\_pi.npy')

        dpidz = median\_interpolation(under, up, dz)

        theta\_0\_avg, theta\_v0\_avg = np.mean(theta\_0), np.mean(theta\_v0)

        g = 9.8

        H -= np.mean(H)

        theta\_c = (dpidz-H)\*theta\_0\_avg\*theta\_v0\_avg/g

        np.save('6-'+z\_name+'\_theta\_c.npy', theta\_c)

    plt.contourf(theta\_c)

    plt.colorbar()

    plt.title(z\_name+" theta\_c'-<theta\_c'>")

    plt.xticks(range(0, 31, 5))

    plt.yticks(range(0, 21, 5))

    plt.savefig('6-'+z\_name+'\_theta\_c')

    #plt.show()

    plt.close()

if \_\_name\_\_ == '\_\_main\_\_':

    for z in range(2, 10, 2):

        f = read\_hw6\_data(z=z, parameter='ff')

        g = read\_hw6\_data(z=z, parameter='gg')

        h = read\_hw6\_data(z=z, parameter='hh')

        theta\_0 = read\_hw6\_data(z=z, parameter='theta0')

        theta\_v0 = read\_hw6\_data(z=z, parameter='thetav0')

        ans\_thetac = read\_hw6\_data(z=z, parameter='ans\_thetac')

        plot\_parameter(f, 'z={} F field(multiplied by {})'.format(str(z), str(10\*\*5)), scale=10\*\*5)

        plot\_parameter(g, 'z={} G field(multiplied by {})'.format(str(z), str(10\*\*5)), scale=10\*\*5)

        plot\_parameter(h, 'z={} H field(multiplied by {})'.format(str(z), str(10\*\*4)), scale=10\*\*4)

        plot\_parameter(theta\_0, 'z={} theta0 field'.format(str(z)), scale=1)

        plot\_parameter(theta\_v0, 'z={} thetav0 field'.format(str(z)), scale=1)

        plot\_parameter(ans\_thetac, 'z={} ans\_thetac field'.format(str(z)), scale=1)

        cal\_pi(f, g, z\_name='z='+str(z))

        f\_under = read\_hw6\_data(z=z-1, parameter='ff')

        g\_under = read\_hw6\_data(z=z-1, parameter='gg')

        cal\_pi(f\_under, g\_under, z\_name='z='+str(z-1))

        f\_up = read\_hw6\_data(z=z+1, parameter='ff')

        g\_up = read\_hw6\_data(z=z+1, parameter='gg')

        cal\_pi(f\_up, g\_up, z\_name='z='+str(z+1))

        cal\_theta\_c(h, theta\_0, theta\_v0, z\_name='z='+str(z))